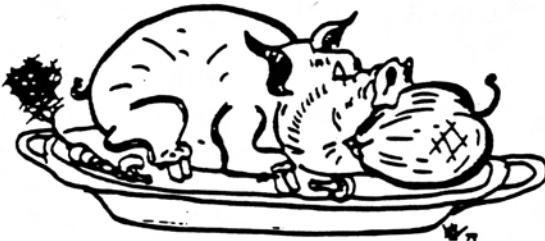


# H. A. A. U. G.



HOUSTON AREA APPLE USERS GROUP

## THE APPLE BARREL

\*\*\* SINGLE COPY PRICE \$1.50 \*\*\*

---

VOLUME 4 NO. 5

SEPT. 1981

President, Bruce Barber

Vice President, Mike Kramer

Published by: A & L ENTERPRISES

---

\*\*\* CONTENTS \*\*\*

Page 1	Club Notes	
Page 2	A Labple?	Terry Sanders
Page 4	Beginners 6502 Assembly Course	Martin Edelstein
Page 6	Keyboard Input On The Apple II+	Mike Kramer
Page 9	Want Ads	
Page 10	Poor Man's Word Processor	Lee Gilbreth
Page 12	"In The News"	D. E. Sanger

Houston Area Apple Users Group  
APPLE BARREL  
Box 42888 #293  
Houston, Texas 77042

<<< CLUB NOTES >>>

The HOUSTON AREA APPLE USERS GROUP is an Apple II user club, not affiliated with Apple, Inc., or with any retail computer store. HAAUG is a member of the International Apple Core and supports its purposes and publications. General membership meetings are held on the second Thursday of each month in the rear chapel of Memorial Lutheran Church, 5600 Westheimer, right by the Jungman Branch Library and west of Chimney Rock. They start at 6:30 p.m. An additional meeting for access to the club software library, problem-solving, and various lectures is held the last Saturday of each month at the University of Texas School of Public Health (in the Medical Center), 6905 Bertner off Holcomb, across from Medical Center entrance #7. We meet on the main floor, first room on the left. Parking is adjacent to the building. These Saturday meetings begin at 2:00 p.m. Bring your Apple if you like!

MEMBERSHIP INFORMATION

Dues are \$18.00 per 12-month period for regular memberships, \$6.00 for students through high school and where no adult member of the family is an Apple user. Please make checks payable to "Houston Area Apple Users Group," and mail to Lee E. Gilbreth, Membership Chair, 3609 Glenmeadow, Rosenberg, TX 77471. This includes a subscription to APPLE BARREL, which is published nine times a year. Newsletter exchanges with similar clubs are invited.

-----\*-----

APPLE BARREL REPRINT POLICY

Unless otherwise indicated within the program or article, any ORIGINAL material published herein may be reprinted without permission by any non-profit Apple club, group or newsletter, PROVIDED proper credit is given to the APPLE BARREL and the article or program author.

-----\*-----

SPECIAL INTEREST GROUPS

OFFICERS / EXECUTIVE BOARD

President	Bruce Barber	469-5805
Vice Pres	Mike Kramer	358-6687
Treasurer	Ray Essig	497-7165
Secretary	Mike Maddock	783-1521
Software	D. Cornwell	774-0671
Hardcopy	Larry Baumann	498-3433
Business	Rudge Allen	622-3979
Membership	LEE Gilbreth	342-2685
Newsletter	Lawrie	469-5805
APPLE HOTLINE	713-668-8685*****	

Members who share a common interest are encouraged to form Special Interest Groups to more fully explore their fields. Meetings may be arranged by common consent of the group and will ordinarily have one member who serves to coordinate or convene the meetings. If you would like to start a group around any given interest, please contact one of the club officers. If you would like to be in touch with others who share one of the following interests with you, please phone the coordinator.

BUSINESS GROUP MEETING

The Business Applications Special Interest Group plans to meet at EBASCO Services 3731 Brairpark on Thursday, 24 September at 7:00 pm to continue their review and discussions of Accounting packages available on the market today. We will be looking at The Comptroler by Apple and any other accounting programs people might want to bring to the meeting.

22#1

## A LABPLE?

Terry Landers  
Department of Biochemistry and  
Molecular Biology  
University of Texas  
Medical School at Houston

An apple in your laboratory can get you into trouble (the "NO FOOD IN LABORATORY AREA" rule), but we have found that an Apple II saves a great deal of trouble (and money) in a number of areas of biochemistry and molecular biology. Our department now has two Apple II's used for tasks ranging from designing genetic engineering experiments to keeping track of grades for 200 medical students.

For genetic engineering and recombinant DNA research, a series of programs allows us to record sequences of DNA molecules (100 to 10,000 bases (letters) long) and then search these sequences for particular smaller sequences. The Apple can produce genetic "maps" from these searches either on the high resolution screen or an XY plotter. We can cut and splice our "electronic molecules" of DNA to create new molecules and generate predicted maps to aid in experimental design. We can also read the genetic code of the DNA to predict the amino acid sequence of proteins that could be produced from the DNA. With a D.C. Hays Micromodem, DNA sequences have been received from large computer-run "sequence banks" in Washington, D.C. and California.

Using a digital voltmeter with BCD output, we record output voltages from an analytical ultracentrifuge, an instrument which allows us to measure how fast molecules move under high gravitational forces. The rate at which a molecule moves is a sensitive measure of its size and shape. Output from the instrument is displayed graphically and the game paddle controls are used to select areas of data for analysis and enter various experimental parameters.

Very rapid chemical and enzyme reactions are also food for our Apples. A digital waveform recorder samples changes in light intensity up to a million times per second and stores the results in its own 2 K byte memory. After the reaction is over, the data is passed to the Apple for further analysis. Again, the analysis programs make good use of the graphical display and paddle input features of the Apple.

Aside from the graphical display capability, one of the most attractive features of the Apple II is the variety of peripheral interface accessory cards available. We use cards with RS-232 (serial), parallel and GPIB (IEEE-488) input and output capabilities to control a printer, a plotter, a spectrophotometer, a digitizer, a Teletype as well as the digital voltmeter and the wave form recorder mentioned above.

# Introducing Memory Management System II

## WHAT IT DOES

**MMS II** is an exciting new software program that can give you an additional 10,700 Bytes of memory to program with.

Imagine. That's an **increase** of almost **30%** to what you now have!! How can you use the extra memory?

- run or write larger programs
- create larger files and arrays
- store more information
- process more data internally to reduce disk access time
- increase the speed of your programs by reducing Apple's housekeeping functions.

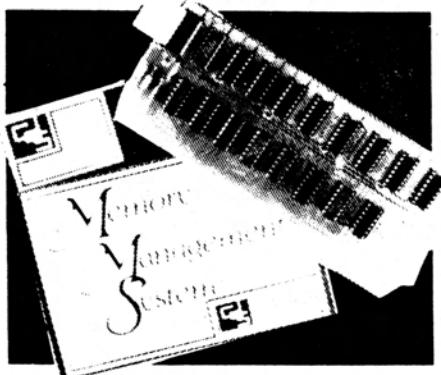
## HOW IT DOES IT

**MMS II** removes the Apple Disk Operating System (DOS) from your Apple's memory and places it on a 16K RAM Expansion board in slot 0. Once DOS is removed, the memory it took up is now **available** for your use. DOS is still active but it no longer takes up precious memory. You have **regained** the full capabilities of your 48K Apple.

## 16K EXPANSION BOARDS

The following 16K Expansion boards are compatible with **MMS II**.

- Apple Language Card       Microsoft RAMCard
- Prometheus MEM-1
- Computer Stop CS16K



- Andromeda 16K RAM Board
- and all others

## DUAL LANGUAGE CAPABILITY

Apple owners who need both Applesoft and Integer BASICs and, simultaneously, wish to have **MMS II** active must have another 16K RAM or ROM card in their computer. **MMS II** will recognize this second card in whatever slot you select.

## ENHANCED VERSION

The original MMS was very well received by many Apple owners. But a few people wrote to us and asked for some modifications. We have incorporated these suggestions into **MMS II**.

- MMS II** is **copyable** and is not protected. It can now become the boot 'Hello' program on your work disk.

- Once **MMS II** is initialized it will **automatically** run whatever program you direct it to.
- MMS II** allows you to specify the slot that contains the second ROM/RAM card.
- All DOS commands are enabled in **MMS II** and are available for your use.
- MMS II** is compatible with programs that use page 3 of memory (i.e. programs that use the ampersand, &, vector.)

## WHAT IS REQUIRED

- 48K Apple II or Apple II Plus
- 1 or more disk drives
- 1 16K RAM Expansion board
- MMS II** by CDS

## UPGRADE POLICY

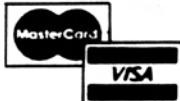
Current owners of MMS can upgrade their system to **MMS II** by returning their original MMS diskette and \$20.00. **MMS II** will be returned immediately.

## HOW TO ORDER

**MMS II** is available through your local computer store, or you can order direct by calling COMPUTER DATA SERVICES at (603) 673-7375. VISA, MasterCard, and COD accepted.

**MMS II**  
\$49.95 (2.00 shipping)

**Andromeda 16K RAM BOARD**  
\$195.00 (3.00 shipping)



P.O. Box 696, Amherst, NH 03031 / (603) 673-7375

Dealer  
Inquiries  
Invited

# BEGINNERS 6502 ASSEMBLY LANGUAGE COURSE FOR THE APPLE II

## PRELIMINARY COURSE OUTLINE:

- 1) LITERATURE
- 2) HEXADECIMAL and BINARY NOTATION
- 3) 6502 INTERNALS
- 4) 6502 EXTERNALS
- 5) the 6502 in the APPLE II
- 6) the 6502 INSTRUCTION SET (pass 1)
- 7) 6502 ADDRESSING MODES
- 8) the 6502 INSTRUCTION SET (one by one)
- 9) ARITHMETIC
- 10) the APPLE MONITOR
- 11) I/O (screen, keyboard, \$C000)
- 12) MORE ARITHMETIC
- 13) DATA TABLES
- 14) the APPLE PERIPHERAL CONNECTOR
- 15) APPLICATIONS

TIME: 10:00 A.M. - 4:00 P.M. (end time is approximate)

DAY: SATURDAY and SUNDAY

DATE: 10/17 - 10/18

PLACE: Southside Place Clubhouse

Garnet Street

Southside Place, tx.

FEE: \$50.00 (preregistration required)

FOR DETAILS CALL MARTIN EDELSTEIN (729-4199)

# NOW! FOR THE FIRST TIME EVER!

The SATURN SYSTEMS 32K board for your Apple II



**TWICE as much memory as existing RAM cards.**

- Run PASCAL, FORTRAN, PILOT and other languages available for the Apple II.
- Compatible with Microsoft CP/M and Z-80 SoftCard.
- Increases Visicalc memory an additional 9K.

**And MORE!**

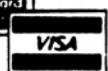
The SATURN SYSTEMS 32K board comes with software that will **automatically** relocate DOS into one of the two 16K banks on the board.

- Imagine! Applesoft and Integer BASIC and DOS capability **AND** 48K to program with.
- Want more? Additional utility software will soon be available that permits you to store BASIC routines, arrays, 'data, and program segments for overlays and chaining.
- And the best part. All this capability is available for **only \$239.00**.

Order your SATURN SYSTEMS 32K board today since quantities are limited!

SSI 32K board  
\$239.00

Shipping (USA)  
3.00



MasterCard

3.00

VISA

3.00

MasterCard

KEYBOARD INPUT ON THE APPLE II+  
BY  
MIKE KRAMER

One of the most important parts of any program is the interaction between the person at the keyboard and the computer. This is frequently referred to as the man/machine interface. In its simplest and least elegant form, this interface will display a list of choices and request a numeric input. The Apple II+ supports several types of keyboard inputs which permit the programmer to include sophisticated input in his programs. This article will attempt to describe some of those techniques which are available in Applesoft.

INPUT COMMAND

The most frequently used keyboard input technique is the INPUT command. When the INPUT command is executed in a program, the program halts execution and waits for input from the keyboard. Consider the following:

100 INPUT A

When this line is executed, a question mark will appear and the program will wait for you to enter a value followed by a RETURN. Multiple items can be entered with a single input statement:

110 INPUT A,B,C

When this line is executed, the program will expect three values separated by commas. If four values are entered, an EXTRA IGNORED message will be displayed. The use of the comma by the Apple as a separator between input items is the reason you cannot directly input a comma as part of an input string variable.

The INPUT statements above leave a great deal to be desired. The main problem is that sometimes the programmer may not want a question mark as part of the input prompt. This is particularly true if a PRINT statement is used for an input prompt:

150 PRINT "ENTER YOUR NAME: ";

160 INPUT ; NAME\$

In this example, "ENTER YOUR NAME: " would be printed on the screen followed by a question mark. The question mark would be on the same line as the prompt statement because the semi-colon at the end of Line 150 would suppress the normal line feed. A shorter way to achieve the same results is:

170 PRINT : INPUT "ENTER YOUR NAME: " ; NAME\$ : PRINT

The PRINT's have been included to provide a line feed before and after the input statement to improve readability on the screen. Cont. Pg. 7

## GET COMMAND

---

The GET command is similar to the INPUT command in that it causes program execution to halt until keyboard entry is made. The most significant difference is that program continues execution as soon as a single character is entered. This is useful when a number of choices can be made by entering a single character. The problem with the GET command is that you don't get a second chance. The following is an example of the use of the GET command for entering a choice:

```
300 PRINT : PRINT "PRESS 'ESC' TO END, 'RET' TO CONTINUE: " : GET A$  
310 IF A$ = CHR$(27) THEN END  
320 IF A$ <> CHR$(13) THEN HOME : GOTO 300  
330 REM PROGRAM CONTINUES
```

The GET command is also useful for inputting any character on the keyboard, including commas and colons. The following subroutine can be used instead of an input command:

```
400 LINE$ = "" : REM NULL STRING  
410 GET CH$ : PRINT CH$ ;  
420 IF CH$ = CHR$(8) AND LEN(LINE$) = 0 GOTO 410  
430 IF CH$ = CHR$(8) THEN LINE$ = LEFT$(LINE$ , LEN(LINE$) - 1) :  
GOTO 410  
440 IF CH$ = CHR$(13) THEN PRINT LINE$ : GOTO 400  
450 LINE$ = LINE$ + CH$ : GOTO 410
```

Line 400 sets LINE\$ to a null string. Line 410 gets a character and prints it on the screen, suppressing the line feed. Line 420 checks for a backspace and no characters on the screen. Line 430 checks for a backspace and drops off the last entered character. Line 440 prints the line on the screen if a RETURN is entered and starts over. Line 450 concatenates the last entered character with the previously entered characters and goes to get another character.

The disadvantage of this input technique is that the Apple may unexpectedly stop to do "garbage collection" while you are touch typing. The ultimate, of course is to write an assembly language program for input, which is beyond the scope of this article and the capabilities of the author.

When using GET while running under Apple DOS, it is possible for the system to "hang" on the next disk access following a GET. This can be avoided by defining "D\$" as follows:

```
500 D$ = CHR$(13) + CHR$(4)
```

Cont. Pg. 8  
PAGE 7

This causes a "RETURN" to be output before each disk access.

#### READING THE KEYBOARD

---

The third input technique permits input through the keyboard "on the fly", without the program pausing. This can be used, for example, to halt a program loop without stopping the program each time through the loop.

When a key is pressed, seven bits (6-0) of information representing the pressed key are stored in memory location \$C000 Hex (-16384 Decimal). Bit 7 is set to indicate that the stored character has not been processed. A program can check to see if a key has been pressed by checking to see if the value stored in memory location -16384 is greater than 128 (bit 7 set). The Apple ASCII character can be determined by subtracting 128 from the contents of location -16384. After obtaining the character, the program should tell the Apple to forget the character it's holding by POKEing a zero into location -16368. The following illustrates how to read the "keyboard on the fly":

```
600 FOR N = 1 TO 10000
```

```
610 CH = PEEK (-16384) : IF CH > 128 THEN POKE -16368,0 : PRINT :  
PRINT "YOU PRESSED THE CHARACTER "; CHR$(CH - 128)
```

```
620 NEXT N
```

#### IDIOT PROOFING INPUT

---

Whichever input technique you decide to use, be sure to take the extra time to assure that your program can survive invalid input data. Always assume the input could potentially bomb your program. For numeric input this involves checking the input for proper range or for specific values:

```
700 INPUT "ENTER CHOICE (1 - 10): "; CH
```

```
710 IF CH < 1 OR CH > 10 THEN PRINT "INVALID INPUT" : GOTO 700
```

```
720 INPUT "WHAT CODE? (12,36,48) "; CO
```

```
730 IF CO <> 12 AND CO <> 36 AND CO <> 48 THEN PRINT "INVALID INPUT"  
: GOTO 720
```

Even though the examples above check for proper numeric values, there is still a booby trap. For example, if the letter "O" is entered instead of the number "0", the program would respond with "?reenter". This problem can be avoided by reading all input variables as strings, whether you are using INPUT or GET. Then take the VAL of the string and check for proper numeric values. Lines 700 and 720 would then be as follows: Cont. Pg. 9

```
700 INPUT "ENTER CHOICE (1 - 10): "; CH$ : CH = VAL(CH$)
```

```
720 INPUT "WHAT CODE? (12,36,48) "; CO$ : CO = VAL(CO$)
```

If a non-numeric value is entered, the VAL evaluates to zero, which is then determined to be invalid.

When entering string variables, you have the choice of accepting "Y" for "YES", or "YUP", or "YEA" or forcing entry of the whole word. The latter is sometimes best if an answer could result in possibly undesired results. An example of string input follows:

```
800 PRINT : INPUT "DO YOU WANT TO WIPE OUT THE FILE? "; ANS$  
810 IF ANS$ = "YES" THEN PRINT D$;"DELETE FILE"  
820 PRINT : INPUT "CONTINUE? "; ANS$ : ANS$ = LEFT$(ANS$,1)  
830 IF ANS$ <> "Y" THEN END  
840 PRINT : PRINT "REALLY? "; : GET ANS : IF ANS$ <> "N" THEN END
```

This has by no means covered the whole subject of how to idiot-proof input, but hopefully it will provide a starting point.

\*\*\*\*\*

FOR SALE.....FOR SALE.....FOR SALE

2 HP 9100B DESK TOP PROGRAMABLE CALCULATORS  
\$1250.00 ea; HP 9800B PROGRAMABLE CALCULATOR  
PRICE NEGOTIABLE: HP 67, LIKE NEW, \$300.00;  
HP 35 - \$35.00; SPERRY UNIVAC DCT 500, 132  
KSR, 110, 115, 300 BAUD \$900.00 - FREE MO-  
DEM W/SERIAL INTERFACE. Harry Haines 493-3088  
\*\*\*\*\*

BASE II 800 BI-DIRECTIONAL PRINTER - READY  
TO RUN. 3 DIFFERENT PRINT TYPES. ALL HARD-  
WARE AND SOFTWARE INCLUDED FOR APPLE: 2 DIA  
PRINTERS - 132 COLUMN KSR. 300 BAUD. INCLDS.  
SOFTWARE. MUST SELL PRICE NEGOTIABLE.\*\*\*\*\*  
Lewis Melton 497-3400\*\*\*\*\*

APPLESOFT FIRMWARE CARD - \$100.00. INCREASES  
USABLE MEMORY AND ALLOWS FREE USE OF EITHER  
INTEGER OR APPLESOFT BASIC ON APPLE II WITH-  
OUT BOOTING A LANGUAGE DISK.\*\*\*\*\*  
Ray Essig 497-7165\*\*\*\*\*

THIS NEWSLETTER HAS BEEN  
PRINTED COURTESY OF\*\*\*\*\*

MOORE BUSINESS SYSTEMS  
1120 Smith Street  
Houston, Texas 77002  
Phone: 237-9063

Deadline for the next  
newsletter...14 Oct 81  
Please mail articles &  
ads to A&L Enterprises  
11803 Rowood Dr  
Houston, TX  
77070

# POOR MAN'S WORD PROCESSER

Back in the April '80 issue of Apple Barrell, Mike Kramer gave us a basic routine for a text processor. In it, a sentence or two could be translated into ASC II code and played out onto a Epson MX-80 printer. This rudimentary program was aptly called "Poor Boy's Word Processor".

Well today, after heavy amendments, I am pleased to announce the boy has reached manhood. Though still "poor", the short program following was written by me to give those who don't own a text processor a chance to play with one.

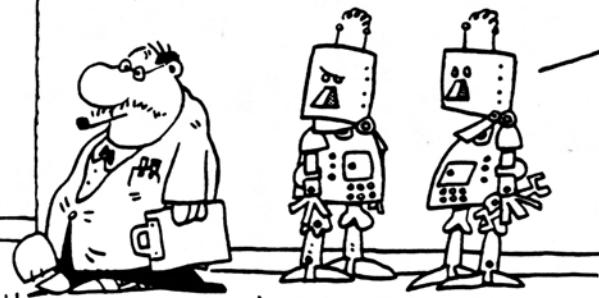
Two things I'd like to see come out of this. One would be for HAAUG members continuing our young man's development (like adding shift-lock for capitals, string find & change, amending files, etc). The other would be using this word processor to create a short article or squib for our Apple Barrell newsletter!

## JLIST

```

1 IF A$ = CHR$(5) THEN 900
5 ONERR GOTO 597
10 DIM LN$(200),LINE$(200):J = 1
    :D$ = CHR$(4): GOTO 1000
21 IF C < > 0 THEN J = C:C = 0:
    GOTO 345
25 LINE$ = "": PRINT "LINE NO. "
    J
30 GET A$: PRINT A$;: NORMAL : IF
    A$ = CHR$(13) THEN 130
51 IF A$ = CHR$(17) THEN END
52 IF A$ = CHR$(4) THEN 300
53 IF A$ = CHR$(12) THEN 1000
54 IF A$ = CHR$(9) THEN 400
55 IF A$ = CHR$(16) THEN 800
56 IF A$ = CHR$(3) THEN 600
57 IF A$ = CHR$(2) THEN 500
58 IF A$ = CHR$(5) THEN 900
59 IF A$ = CHR$(19) THEN 700
60 IF A$ = CHR$(8) AND LEN (L
    INE$) < = 1 THEN 25
70 IF A$ = CHR$(8) AND LEN (L
    INE$) > 1 THEN LINE$ = LEFT$(
        LINE$, LEN (LINE$) - 1): GOTO
    30
80 LINE$ = LINE$ + A$: IF LEN (L
    INE$) > 74 THEN PRINT CHR$(
    ?)
90 IF A$ = CHR$(27) THEN INVERSE
100 GOTO 30
130 LINE$(J) = LINE$; IF LEN (LI
    NE$) = 0 THEN PRINT : GOTO
    250
170 CAP = 32: FOR I = 1 TO LEN (
    LINE$):CH$ = MID$(LINE$,I,
    1): IF ASC (CH$) = 27 THEN
    CAP = 0: GOTO 240
210 IF ASC (CH$) < 65 OR ASC (
    CH$) > 90 THEN CAP = 0
220 LTR$ = CHR$ (ASC (CH$) + CA
    P):LN$ = LN$ + LTR$:CAP = 32
240 NEXT
250 LN$(J) = LN$;LN$ = ""
260 J = J +(1: GOTO 21
300 HOME : VTAB 5: PRINT J - 1"*
    LINES OF TEXT IN THIS FILE."
    : PRINT : INPUT "START AT LI
    NE NO. ?";A: INPUT "AND END
    WITH LINE NO. ?";B: HOME : IF
    A = 0 OR B = 0 THEN 300
305 FOR K = A TO B: PRINT "LINE
    NO. ";K:LINE$ = LINE$(K): IF
    LEN (LINE$) = 0 THEN PRINT
    : GOTO 340
310 FOR I = 1 TO LEN (LINE$):CH
    $ = MID$(LINE$,I,1): IF CH
    $ = CHR$(27) THEN INVERSE
    : GOTO 330 Cont. Pg. 11

```



YEAH, THEY'RE  
PRETTY UGLY,  
BUT THEY SURE  
KNOW A LOT  
ABOUT SOFTWARE.

```

320 PRINT CH$;: NORMAL
330 NEXT I: PRINT
340 NEXT K: IF A$ = CHR$(3) THEN
    PRINT "NEW ";: J = K - 1: GOTO
    25
345 FOR I = 1 TO 40: PRINT "-";:
    NEXT : GOTO 25
400 INPUT "INSERT BEFORE WHICH L
INE? "; K: FOR I = J TO K + 1
    STEP - 1: LINE$(I) = LINE$(I - 1): LN$(I) = LN$(I - 1): NEXT
    : C = J + 1: J = K: GOTO 25
500 PRINT "NAME OF OLD FILE?": INPUT
    "(TYPE '0' FOR CATALOG)": N$.
    : IF N$ = CHR$(48) THEN PRINT
    D$"CATALOG": PRINT : GOTO 50
    0
505 ONERR GOTO 597
510 PRINT D$"OPEN"N$.
520 PRINT D$"READ"N$.
530 INPUT J: FOR I = 1 TO J: INPUT
    LN$(I): NEXT
540 PRINT D$"CLOSE": FOR K = 1 TO
    J: LN$ = LN$(K): IF LEN (LN$)
    > = 0 THEN 595
560 FOR I = 1 TO LEN (LN$): CH$ =
    MID$(LN$, I, 1): IF ASC (CH
    $) < 65 THEN LTR$ = CH$: GOTO
    590
570 IF ASC (CH$) > = 65 AND ASC
    (CH$) < = 90 THEN LTR$ = CHR$(
    27) + CH$: GOTO 590
580 LTR$ = CHR$ (ASC (CH$) - 32
    )
590 LINE$ = LINE$ + LTR$: NEXT I:
    LINE$(K) = LINE$; LINE$ = ""
595 NEXT K: J = J + 1: GOTO 300
597 POKE 216, 0: PRINT D$"CLOSE":
    PRINT "FILE NOT FOUND": PRINT
    D$"DELETE"N$: GOTO 345
600 INPUT "CHANGE WHAT LINE ?": A
    : B = A: C = J: PRINT "OLD ":
    GOTO 305
700 INPUT "NAME OF NEW FILE": N$.
    : PRINT D$"OPEN"N$.
720 PRINT D$"WRITE"N$.
730 PRINT J - 1: FOR I = 1 TO J - 1:
    PRINT LN$(I): NEXT
770 PRINT D$"CLOSE": GOTO 345
800 PRINT D$"PR#1"
805 PRINT CHR$(9) "80N"
810 FOR I = 1 TO J - 1: PRINT LN
    $(I): NEXT
840 PRINT CHR$(9) "I"
850 PRINT D$"PR#0": GOTO 345

```

```

900 INPUT "RUB OUT WHICH LINE ?"
    : K: FOR I = K TO J - 1: LINE$(
    I) = LINE$(I + 1): LN$(I) =
    LN$(I + 1): NEXT : J = J - 1:
    GOTO 345
1000 HOME : HTAB 7: PRINT "POOR
    MAN'S WORD PROCESSER": HTAB
    7: PRINT "-----"
1010 PRINT : PRINT "CONTROL B =
    BRING IN OLD FILE": PRINT : PRINT
    "CONTROL D = DISPLAY THIS FI
    LE"
1015 PRINT : PRINT "CONTROL I =
    INSERT NEW LINE": PRINT : PRINT
    "CONTROL C = CHANGE A LINE"
1017 PRINT : PRINT "CONTROL E =
    ERASE A LINE"
1020 PRINT : PRINT "CONTROL S =
    SAVE THIS FILE TO DISK": PRINT
    : PRINT "CONTROL P = PRINT T
    HIS FILE"
1040 PRINT : PRINT "CONTROL L =
    LIST THIS MENU": PRINT : PRINT
    "CONTROL Q = QUIT THE PROGRA
    M"
1060 PRINT : PRINT "ESCAPE BEFOR
    E A LETTER = CAPITAL LETTER.
    "; VTAB 23: INPUT "<RETURN>" :
    A$: HOME : GOTO 25

```

-LEE GILBERTH

WANT TO SNEAK-IN AN ILLEGAL COMMAND?

>LIST

```

    0 POKE 74, 9999 MOD 256: POKE
    75, 9999/256: POKE 204, PEEK
    (74): POKE 205, PEEK (75)
    10 REM
    20 REM ABOVE LINE "0" SETS
        LOMEM TO "9999"
    30 REM
    40 REM (NOTE: ONLY WAY CMD CAN
        BE MADE IN INTEGER)

```

## Baffled Users Of Computers Get Some Help

By DAVID E. SANGER

Staff Reporter of THE WALL STREET JOURNAL

BOSTON—Sold on personal computers but baffled by their technical fine points, businessmen around the country are joining "user groups" to help cope with the machines.

The cooperative groups are trying to succeed where manufacturers largely have failed, in holding the hands of first-time computer owners and helping them rout the myriad bugs and mysteries that plague new systems. Although nobody knows exactly how many user groups exist, people in the computer business put the number at more than 1,000 ranging in size from three members to hundreds.

For many members, a user group offers the only useful source for learning how to turn \$2,000-to-\$8,000 worth of electronic hardware into something more than a piece of furniture. A typical group, the Boston Computer Society, offers regular seminars on such topics as program design and matching hardware, which won't work if mismatched. Some businessmen drop by the society's downtown display room at lunch to try out equipment installed by several manufacturers.

"If people want to find out how to make their computer play music, or customize packages, or talk, they find others in the user groups very helpful," says Jonathan Rotenberg, a sophomore at Brown University and president of the Boston society. Each of the society's 2,200 members belongs also to a specific subgroup of owners of the same kind of equipment.

User groups were different in the early days of personal computers, when most buyers were hobbyists dedicated to mastering the machines on their own. "The whole idea of taking the machine out of the box and having it work the first time ruined their whole day," Garland Asher, director of financial planning for Tandy Corp., jokes. Tandy's Radio Shack division sells the popular TRS 80 line of personal computers.

Buyers now, however, are mainly businessmen, financial analysts, doctors and other professionals who need the machines for specific application rather than general tinkering. Programming at its easiest is tedious and complicated, and most owner manuals barely hint at the power and versatility available in a machine to the skilled user.

The mainstays of the Boston society's membership are professionals who pay \$15 a year for access to the expertise of their fellow members. Dr. Randall M. Sneider, a Newton dentist, used club help in designing

PAGE 23  
FRIDAY, JULY 10, 1981

## Many Baffled Users Of Computers Getting Some Needed Help

*Continued From Page 23*

the software needed to cut through a mountain of dental and office paperwork with his Apple computer. Several financial analysts and accountants have got help in constructing computer financial models.

Despite the market potential in user groups, most manufacturers have dealt with them cautiously or even avoided them. For one thing, most manufacturers have preoccupied themselves with expansion. For another, some members of user groups have illegally duplicated copyrighted programming, cutting into the companies' own sales. "We felt we did not want to endure this kind

of software piracy," says Charles A. Phillips, senior vice president of Radio Shack, explaining his company's aloofness from the groups.

But Radio Shack and Apple Computer, a leading competitor, are scrambling to better their relations with consumers. Apple has even created an "International Apple Core" service to coordinate exchanges between user groups. Some Apple owners still are less than happy with Apple's efforts. "We would like the names of some specific people we could call at Apple with technical questions," says Dick Hubert, president of the Apple Puget Sound Library Exchange in Washington State. Apple does have a telephone hotline for users, but Apple managers worry that the company might be flooded by requests for assistance.

Many clubs sell software written by members. In Denver, Douglas M. Borwick, who started a group in February for users of Intertec Data Systems equipment, already has seven employees, office space and a healthy retail-wholesale business in software and peripheral equipment. At Intertec, that's fine. "Our main objective is to sell hardware," says Donald Wells, executive vice president, "and if an individual group can help us sell hardware, then we're all for them."



**MOORE  
BUSINESS  
CENTER**

Business Forms • Computer Supplies • Business Equipment • Instant Printing

10% DISCOUNT TO HAAUG MEMBERS  
(Discount does not apply on current label sale.)

## **IF YOU WANT TO INCREASE YOUR BANKROLL . . . THEN STICK 'EM UP . . .**

**with Avery Labels, that is.**

NOW THROUGH OCTOBER 30, COME IN AND STICK US UP FOR BIG SAVINGS ON Avery COPIER AND PIN FED ADDRESS LABELS! OUR LIST PRICE IS BELOW THE MANUFACTURER'S SUGGESTED PRICE, AND NOW WE'RE TAKING OFF AN ADDITIONAL 20%.

COPIER AND CONTINUOUS ADDRESS LABELS SAVE YOU HOURS OF TEDIOUS WORK IN PREPARING MAILINGS, AND NOW YOU CAN ALSO SAVE LOTS OF PRECIOUS DOLLARS IN BUYING THEM.

SEE THE CHART BELOW FOR A COMPARISON OF SUGGESTED, REGULAR AND SALE PRICES.

<u>Product Number</u>	PRICES PER BOX:		STICK-'EM-UP SALE PRICE	Description
Copier Labels	Manufacturer's Suggested Price	Our Regular Price		
5351 (3300 labels/box)	\$21.20	\$16.75	\$13.40	Dry Toner
5331 (3000 labels/box) (Both - 100 sheets/box)	21.20	16.75	13.40	Liquid Toner
Pin Fed Address Labels				
4139 (1000 qty.)	\$ 8.60	\$ 7.39	\$ 5.91	3½ x 15/16
4140 (3000 qty.)	19.30	14.99	11.99	3½ x 15/16
4013 (5000 qty.)	25.45	19.79	15.83	3½ x 15/16
4141 (1000 qty.)	\$13.25	\$10.95	\$ 8.76	4 x 1 7/16
4142 (3000 qty.)	31.00	23.99	19.19	4 x 1 7/16
4014 (5000 qty.)	43.50	29.95	23.96	4 x 1 7/16

COME SEE US FOR COMPUTER SUPPLIES, OFFICE SUPPLIES, COPYING, AND OFFSET PRINTING

**1120 Smith Street / Houston, TX 77002 / (713) 237-9063**

An Equal Opportunity Employer

THE PHRASER  
SPEECH SYNTHESIZER

- \* VOTRAX SC-01A SPEECH SYNTHESIZER CHIP
- \* SAY ANY ENGLISH WORD OR PHRASE
- \* CREATE SOUND EFFECTS
- \* 16 COMPUTER CONTROLLED VOLUME LEVELS
- \* 16 COMPUTER CONTROLLED PITCH LEVELS
- \* 3 1/2" SPEAKER
- \* ATTRACTIVE ENCLOSURE (table or wall mount)
- \* PLUGS INTO THE GAME I/O SOCKET
- \* 12 V. CALCULATOR POWER SUPPLY (provided)
- \* THE PHRASER STARTER DICTIONARY (500 words)
- \* WORDMAKER: Utility program to add words to your PHRASER dictionary
- \* PHRASE ASSEMBLER: Utility to create word/phrase tables from the PHRASER dictionary for use in your applications
- \* DEMO PROGRAMS (education, games, etc.)

INTRODUCTORY PRICE \*\*\* \$225.00 \*\*\*

MODEL SS-01B for the APPLE II OR II+

MODEL SS-01A for the ATARI 400 OR 800

GEMMAE PRODUCTS INC.  
10935 Cedarhurst Drive  
Houston, Tx. 77096  
(713) 729-4199

Houston Area Apple Users Group

APPLE BARREL

A & L ENTERPRISES, Publishers

11803 Rowood Drive

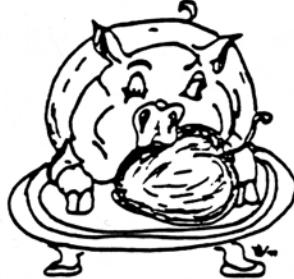
Houston, TX 77070

Postmasters:

Address Correction Requested:

Forwarding and Return Postage Guaranteed

BULK RATE  
U.S. POSTAGE  
PAID  
HOUSTON, TEXAS  
PERMIT 3936



H.A.A.U.G